"*Express Mail*" mailing label number: EV 339774319 US

Date of Deposit: September 30, 2003

INFINEON PATENT
Case No. 2002 – P16865US

BRINKS HOFER PATENT
Case No. 10808/95

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTOR: RAINER HOEHLER

TITLE: SELECTIVE ADDRESS-RANGE REFRESH

ATTORNEYS:

John C. Freeman
Reg. No. 34,483
BRINKS HOFER
GILSON & LIONE
P.O. Box 10395
Chicago, Illinois 60610
(312) 321-4200

SELECTIVE ADDRESS-RANGE REFRESH

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to the field of memory systems, and in particular memory systems that employ a refresh operation.

Discussion of Related Art

[0002] It is well known in the art that various types of personal computers, such as desktop computers and battery-operated notebook computers, include a central processing unit (CPU) and a main memory to which the central processing unit accesses. The central processing unit executes programs loaded on the main memory, and sequentially writes the results obtained by program execution into work areas in banks of the main memory so that the computer processing is performed.

[0003] The main memory is composed of a random access memory (RAM), such as SRAM (static RAM) and DRAM (dynamic RAM). For the main memory, DRAM is generally used because DRAM has a simple cell structure and is cheaper. Accordingly, the discussion to follow will concentrate on known DRAM memory systems.

[0004] DRAM memory cells in the main memory are arranged as a matrix. In order to address memory cells individually, first, an Activate Command is issued with a row address, and then, read or write commands are issued with the column address. In the DRAM memory cells, data are stored as electric charges on a capacitor. Thus, when data are written to the memory cells and are left for an extended period of time, the charges leak from the capacitor

and the stored data are lost. To prevent such data loss, the written data needs to be refreshed/rewritten at predetermined time intervals.

[0005] Known refresh operations include accessing a specific memory cell row to refresh all of the cells along that row. In order to refresh all of the row addresses, a refresh address counter is required that designates refresh addresses sequentially. In addition, the known refresh operations provide either a refresh cycle or issues a refresh request at a predetermined period of time.

[0006] One known method to refresh the memory contents is to serially access all rows with an activate – precharge command-sequence. For this method, a refresh address counter designates refresh row addresses that must be provided from outside the memory.

[0007] A second known refresh operation is generally known as autorefresh where a refresh request is supplied to the memory by sending an Autorefresh command. The refresh addresses are generated by an address counter within the DRAM such that no external address counter is required.

[0008] A third known refresh operation is self-refresh, which allows the data in the DRAM to be refreshed even while the rest of the system is powered down. During self-refresh an internal timing circuit and an internal address counter generate the refresh operations for all rows in time intervals sufficiently short to keep the stored data intact. This allows for very low power consumption since the time-intervals between refreshes can be optimized and all other circuits can be powered down.

[0009] FIG. 1 is a schematic diagram illustrating the arrangement of a known computer system 100 that has both the normal refresh function and the self-refresh function. A DRAM device 102, including a DRAM array 103, and a memory controller unit 104 are connected to each other by a bus 106 and an I/O device 108. Outside the DRAM device 102 are provided a normal refresh circuit 110, which forms a part of the memory controller unit 104 that performs a refresh operation while the memory controller unit 104 is accessing the memory, and a global clock 112. Inside the DRAM device 102 are provided a self-refresh circuit 114 that performs a relatively slow refresh operation, and an internal timing generator 116 that supplies a relatively long interval signal to the self-refresh circuit 114. In addition, a switch 118 is provided to select either the normal refresh circuit 110 or the self-refresh circuit 114 for refreshing the DRAM device 102.

[0010] One disadvantage of such a system is that all address rows of the DRAM 103 are self-refreshed even if some address rows do not contain data. Accordingly, there is a waste in power used during self-refreshing

SUMMARY OF THE INVENTION

[0011] One aspect of the present invention regards a memory control system that includes a memory controller and a memory device connected to the memory controller via a command bus, wherein command signals are directed from the memory controller to the memory device. The memory device includes one or more memory banks, a row address register and a command decoder that is connected to the row address register and receives the command signals and controls the contents of the row address register. A

refresh circuit is connected to the one or more memory banks and the row address register, wherein the refresh circuit avoids unnecessary power consumption for refreshing the one or more memory banks.

[0012] A second aspect of the present invention regards a memory control system including a memory controller, a memory device connected to the memory controller via a command bus, wherein command signals are directed from the memory controller to the memory device. The memory device includes one or more memory banks, a first row address register, a second row address register and a command decoder that is connected to the first row address register and the second row address register and receives the command signals and controls the contents of the first row address register and the second row address register. A refresh circuit connected to the one or more memory banks and the row address register, wherein the refresh circuit avoids unnecessary power consumption for refreshing the one or more memory banks.

[0013] A third aspect of the present invention regards a memory control system including a memory controller, a memory device connected to the memory controller via a command bus, wherein command signals are directed from the memory controller to the memory device. The memory device includes one or more memory banks, one row address register for each of the one or more memory banks and a command decoder that is connected to each of the row address registers and receives the command signals and controls the contents of the row address registers. A refresh circuit connected to the one or more memory banks and the row address

registers, wherein the refresh circuit avoids unnecessary power consumption for refreshing the one or more memory banks.

[0014] A fourth aspect of the present invention regards a method of refreshing one or more memory banks of a memory device that receives command signals from a memory controller. The method including monitoring command signals received by a memory device and refreshing the one or more memory banks based on the monitored command signals so as to avoid unnecessary power consumption for refreshing the one or more memory banks.

[0015] Each of the above aspects of the present invention provides the advantage of reducing power

[0016] The present invention, together with attendant objects and advantages, will be best understood with reference to the detailed description below in connection with the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 schematically shows an embodiment of a known memory device that includes normal and self-refresh circuits;

[0018] FIG. 2 schematically shows a first embodiment of a memory system in accordance with the present invention;

[0019] FIG. 3 shows a flow chart of a first embodiment of a refresh process in accordance with the present invention to be used with the memory system of FIG. 2;

[0020] FIG. 4 schematically shows a second embodiment of a memory system in accordance with the present invention;

[0021] FIG. 5 shows a flow chart of a second embodiment of a refresh process in accordance with the present invention to be used with the memory system of FIG. 4;

[0022] FIG. 6 schematically shows a third embodiment of a memory system in accordance with the present invention;

[0023] FIG. 7 shows a flow chart of a third embodiment of a refresh process in accordance with the present invention to be used with the memory system of FIG. 6;

[0024] FIG. 8 schematically shows a fourth embodiment of a memory system in accordance with the present invention;

[0025] FIG. 9 shows a flow chart of a fourth embodiment of a refresh process in accordance with the present invention to be used with the memory system of FIG. 8;

[0026] FIG. 10 schematically shows a fifth embodiment of a memory system in accordance with the present invention;

[0027] FIG. 11 shows a flow chart of a fifth embodiment of a refresh process in accordance with the present invention to be used with the memory system of FIG. 10;

[0028] FIG. 12 schematically shows a sixth embodiment of a memory system in accordance with the present invention; and

[0029] FIG. 13 shows a flow chart of a sixth embodiment of a refresh process in accordance with the present invention to be used with the memory system of FIG. 4;

DETAILED DESCRIPTION OF THE INVENTION

[0030] The present invention is best understood by a review of the embodiments and modes of operation represented by FIGS. 2-13. As shown in FIG. 2, a memory system 200 includes a memory controller 202 and a memory device 204. The memory controller 202 executes memory accesses (including both read accesses and write accesses) of the memory device 204 in response to memory access requests issued by a central processing unit (not illustrated).

[0031] The memory controller 202 and the memory device 204 are connected together by a command bus 205 of command signals, an address bus 207 of address signals, and a data bus 209 of data signals, clock signals (not illustrated) and datastrobe signals (not illustrated).

[0032] The memory controller 202 has a normal refresh circuit 206 that performs a normal refresh operation in a manner similar to that described previously. The normal refresh circuit 206 provides a normal refresh cycle every predetermined interval, by sending an autorefresh signal to the memory device 204 through the command bus 205. Incorporated in the memory device 204 is a self-refresh circuit 210, a self-refresh timer 214, a command decoder 216, and a maximum row address register 212. The self-refresh circuit 210 includes a refresh address counter for incrementing a row address to be refreshed at each refresh cycle, and a controller for, in response to a refresh request, controlling access to a row address such that all rows get refreshed within a given time period to avoid loss of memory contents. The address counter covers all row-addresses and restarts at the lowest row

address after the highest row-address has been refreshed. The incorporated refresh circuit 210 executes either a "normal refresh" operation and a "self-refresh" operation upon receipt of a corresponding refresh request. A normal refresh operation is realized by responding to a memory autorefresh refresh request from the normal refresh circuit 206, and by accessing a designated row address.

[0033] A self-refresh operation is begun when the memory device 204 is put into self-refresh mode through the respective command sequence from the memory controller 200. In the self-refresh mode, a self-refresh request is issued to the self-refresh circuit 210 from the memory controller 200 every predetermined time period that is triggered by a self-refresh timer 214. In state-of-the art memories devices, such a self-refresh circuit leads to an activate / precharge sequence for all memory banks 208 of the memory device 204 in parallel. An example of a known self-refresh circuit that can be adapted for use with the present invention is the 256Mbit-DDR-SDRAM manufactured and sold by Infineon under the part number HYB25D25616OBT-6.

[0034] As shown in FIG. 2, the memory device 204 includes a maximum row address register 212 that is in communication with both a command decoder 216 and the self-refresh circuit 210. It is a register which stores a single number ranging from the lowest (usually 0) to the highest row-address of the memory banks 208.

[0035] The command decoder monitors all write commands directed to the memory banks 208 and controls the contents of the maximum row address

register 212. In the discussion to follow, the contents of the maximum row address register 212 will be denoted as ROW_{max}, a digital value which can cover the same range as the row-address space of the memory device 204.

[0036] The digital value of ROW_{max} of the maximum row address register 212 is initially set to zero during the power-up sequence of the memory device 204. Afterwards, the value of ROW_{max} of the maximum row address register 212 is controlled by the command decoder 216 of the memory device 204. Whenever a write-command is issued to the memory device 204, the command decoder 216 decodes this write command and also decodes the row-address to which data are written. Whenever the decoded row-address is higher than the value of ROW_{max} presently stored in the maximum row address register 212, then the value of ROW_{max} is updated to have a value that corresponds to the presently decoded row-address.

[0037] In the case when memory contents are not relevant for the system anymore and thus no refreshing is required anymore, a command sequence, usually defined as an extended mode register set can be used to program/reset the value of ROW_{max} of the maximum row address register 212 to zero or with an alternative implementation to any programmed value .

[0038] The self-refresh circuit 210 monitors the contents of the maximum row address register 212 and starts the activate/precharge sequence only for those rows with an address lower or equal than the stored value of ROW_{max} of the maximum row address register 212. For those rows with addresses that have a value that is greater than the stored value of ROW_{max}, the self-refresh circuit 210 suppresses the refresh of wordlines of those rows. Thus, the

circuit 210 avoids unnecessary power consumption for refreshing rows which are defined to not require to be refreshed by the maximum row address register 212. The circuit 210 can be altered in such a way that it either checks the maximum row address register 212 1) only in case of self-refresh mode or 2) both in self-refresh and auto-refresh mode.

[0039] As shown in FIG. 3, two parallel processes performed by decoder 216 are controlling the self-refresh - and with respective implementation also autorefresh - process 300. The two processes are performed by a global control circuit that includes the command decoder 216. The sub-process 302 controls the contents of the maximum row address register 212.

[0040] As shown in FIG. 3, the self-refresh process 300 involves at power up (step 304) resetting the value ROW_{max} stored in the maximum-row address register 212 to zero per step 306. Next, the row-addresses of commands sent to the memory banks 208 are analyzed in the manner set forth below. In particular, detection of an extended mode register access for the maximum row address register 212 is performed per step 308. If extended mode register access is detected per step 308, then the value of ROW_{max} of the maximum row address register 212 is reset to the value 0 (step 310) or alternatively programmed to any other value (not illustrated) so as to declare the contents of the particular bank in question irrelevant. If an extended mode register access is not detected per step 308, then the value of ROW_{max} is not altered.

[0041] Irrespective of whether or not an extended mode register access is detected per step 308, the presence of a write command is determined per

step 312. If a write command is detected, then the row-address i gets decoded by command decoder 216 per step 314. If no write command is decoded then the process returns to step 308, as shown in FIG. 3, where another command sent to the memory banks 208 is monitored.

[0042] If a write command is detected, a comparison between the decoded value i and the value of ROW_{max} is performed per step 316. In particular, if the decoded value i of the row-address is greater than the value ROW_{max}, the value ROW_{max} is set to the decoded value i and stored in the maximum row address register per step 318. Then the process is repeated starting at step 308 where another command sent to the memory banks 208 is monitored.

[0043] Note that if the decoded value i is less than or equal to the value ROW_{max}, per step 316, then the above-described process is repeated at step 308 where another command sent to the memory banks 208 is monitored.

[0044] Note that that the addresses monitored at step 308 in the process described above are not done sequentially with respect to each memory bank 208. An alternative manner of determining the maximum row address is to determine the maximum row address for each memory block 208 separately and using the greatest value of the determined set of maximum row addresses as the value of ROW_{max}.

[0045] The subprocess 320 shows the behavior during self-refresh and autorefresh requests. In particular, the command decoder 216 detects whenever a self-refresh or auto-refresh request for any row-address occurs per step 322. If no self-refresh or autorefresh is detected per step 322, then

the process is then repeated and returns to step 322. If a self-refresh or autorefresh request is detected, then the value of the requested row-address for the refresh is compared with the value of ROW_{max} per step 324.

[0046] If the value of the requested row-address is lower or equal to the value ROW_{max} of the maximum row address register, then the rows of all banks 208 with this row-address get refreshed per step 326. The process is then repeated and returns to step 322.

[0047] Note that should the value of the requested row-address be determined in step 324 to be higher than the value ROW_{max}, then the refresh request gets ignored and the process returns to step 322, which results in lower power consumption for the refresh.

[0048] An alternative embodiment of a self-refresh memory device and corresponding self-refresh process are shown in FIGS. 4-5, wherein components and processes similar to those shown in FIGS. 2-3 are identified with like numerals. In particular, a memory system 400 includes a memory controller 202 and a memory device 404. The memory controller 202 has a normal refresh circuit 206 that performs a normal refresh operation in a manner similar to that described previously as explained previously with respect to FIGS. 2-3.

[0049] The memory controller 202 and the memory device 404 are connected together by a bus 205 of command signals, a bus 207 of address signals, a bus 209 of data signals, clock signals (not illustrated) and datastrobe signals (not illustrated) in a manner similar to that described previously with respect to FIGS. 2-3.

[0050] The memory device 404 has memory banks 208. Different from the memory device 204 in FIG. 2, the memory device 404 in this kind of implementation includes one maximum row address register 412 per memory bank 208. It also includes refresh enable circuits 418, one for each memory bank 208.

[0051] The command decoder 416 monitors all write commands directed to the memory banks 208 and controls the contents of the maximum row address registers 412. In the discussion to follow, each bank address will be denoted as b with b=1,.....n with n denoting the total number of memory banks 208. The contents of the maximum row address register, connected to bank b will be denoted as $ROW_{max,b}$, a digital value which can cover the same range as the row-address space of the memory device.

[0052] The contents $ROW_{max,\,b}$ (b=1, 2, . . . n) of all maximum row address are set to zero at the power-up sequence of the memory device 404. Afterwards, the contents of the maximum row address registers 412 are controlled by the command decoder 416 of the memory device 404. Whenever a write-command is issued to the memory device 404, the command decoder 416 decodes this write command and also decodes the bank address b and the row-address to which data are written. The decoder 416 then compares the value $ROW_{max.\,b}$ of the maximum row address register corresponding to the bank b and the row-address of the decoded write command. If the row-address of the decoded write command is greater than the value $ROW_{max,\,b}$ of the maximum row address register 412 of the addressed bank b, the value $ROW_{max,\,b}$ of the maximum row address register 412 for bank b is changed to

value of the row-address of the decoded write command. A command sequence – usually defined as an extended mode register set - can be used to reset the maximum row address registers 412 of all banks 208 to zero. . Alternatively, an implementation is possible where each maximum row address register is reset individually with an extended mode register. Another implementation can allow setting the contents of all maximum row address registers to a programmable value. Another implementation allows setting the contents of each maximum row address register to a programmable value individually.

[0053] The self-refresh circuit 410 shown in FIG. 4 sends refresh requests, including the row-address information, to refresh enable circuits 418 for all banks 208. These refresh enable circuits 418 monitor the contents of the maximum row address registers for a corresponding memory bank 208. Only if the row-address of the refresh request from the self-refresh circuit 410 is lower or equal to the value of $ROW_{max,b}$ of the maximum row-address register 412 of a bank b, a refresh sequence is started for bank b. The circuit 410 can be altered in such a way that it either does check the maximum row address register only in case of self-refresh mode or both in self-refresh and auto-refresh mode.

[0054] As shown in FIG. 5, the self-refresh process 500 involves at power up (step 504) resetting the value $ROW_{max,\,b}$ stored in the maximum-row address registers 412 for each memory bank b (b=1, 2, . . . n) to zero per step 506. Next, the row-addresses of commands sent to the memory banks 208 are analyzed in the manner set forth below. In particular, detection of an

extended mode register access for each of the maximum row address registers 412 is performed per step 508. If extended mode register access for a particular bank b is detected per step 508, then the value of $ROW_{max, b}$ of the maximum row address register 412 for addressed bank b is reset to the value 0 (step 510) or alternatively programmed to any other value (not illustrated) so as to declare the contents of the particular bank in question irrelevant. If an extended mode register access is not detected per step 508, then the value of $ROW_{max, b}$ is not altered.

[0055] Irrespective of whether or not an extended mode register access is detected per step 508, the presence of a write command is determined per step 512. If a write command is detected, then the row-address i and addressed bank b gets decoded by command decoder 416 per step 514. If no write command is decoded then the process returns to step 508, as shown in FIG. 5, where another command sent to the memory banks 208 is monitored.

[0056] If a write command is detected, the bank address b is decoded and a comparison between the decoded value i of the row-address and the value of $ROW_{max, b}$ is performed per step 516. In particular, if the decoded value i of the row-address is greater than the value $ROW_{max, b}$, the value $ROW_{max, b}$ is set to the decoded value i and stored in the maximum row address register 412 corresponding to bank b per step 518. Then the process is repeated starting at step 508 where another command sent to the memory banks 208 is monitored.

[0057] Note that if the decoded value i is less than or equal to the value

$ROW_{max, b}$, per step 316, then the above-described process is repeated at step 508 where another command sent to the memory banks 208 is monitored.

[0058] The subprocess 520 shows the behavior during self-refresh and autorefresh requests. In particular, the command decoder 416 detects whenever a self-refresh or auto-refresh request for any row-address of a particular bank b occurs per step 522. If no self-refresh or autorefresh is detected per step 522, then the process is then repeated and returns to step 522. If a self-refresh or autorefresh request is detected, then several steps are started in parallel, one for each bank 208. In particular, the value of the requested row-address for the refresh is compared by the refresh enable circuits 418 in parallel with each of the values of $ROW_{max, b}$ of the banks 1, 2, . . . b per step 524.

[0059] If the value of the requested row-address is lower or equal to the value $ROW_{max, b}$ of the maximum row address register 412 for bank b, then bank b gets refreshed at this row-address of the refresh request per step 526. The process is then repeated and returns to step 522.

[0060] Note that should the value of the requested row-address be determined in step 524 to be higher than the value $ROW_{max, b}$,then the refresh request for bank b of the requested row-address is ignored and the process returns to step 522, which results in lower power consumption for the refresh.

[0061] An alternative variation of the self-refresh memory device and corresponding self-refresh process of FIGS. 2-3 are shown in FIGS. 6-7,

wherein components and processes similar to those shown in FIGS. 2-3 are identified with like numerals. As shown in FIG. 6, the memory device 604 includes a minimum row address register 612 that is in communication with both the command decoder 616 and the self-refresh circuit 610. It is a register which stores a single number that is as low as the lowest row-address of the memory banks 208.

[0062] The command decoder 616 monitors all write commands directed to the memory banks 208 and controls the contents of the minimum row address register 612. In the discussion to follow, the contents of the minimum row address register 612 will be denoted as ROW_{min}, a digital value which can cover the same range as the row-address space of the memory device 604.

[0063] The digital value of ROW_{min} of the minimum row address register 612 is initially set to a maximum row address value during the power-up sequence of the memory device 604. Afterwards, the value of ROW_{min} of the minimum row address register 612 is controlled by the command decoder 616 of the memory device 604. Whenever a write-command is issued to the memory device 604, the command decoder 616 decodes this write command and also decodes the row-address to which data are written. Whenever the decoded row-address is lower than the value of ROW_{min} presently stored in the minimum row address register 612, then the value of ROW_{min} is updated to have a value that corresponds to the presently decoded row-address. A command sequence– usually defined as an extended mode register set can be used to program/reset the value of ROW_{min} of the minimum row address register 612 to the maximum row address value or with an alternative

implementation to any programmed value so as to declare the contents of the bank in question irrelevant.

[0064]The self-refresh circuit 610 monitors the contents of the minimum row address register 612 and starts the activate/precharge sequence only for those rows with an address that has a value that is greater than or equal to the stored value of ROW_{min}. For those rows with addresses that have a value that is less than the stored value of ROW_{min}, the self-refresh circuit 610 suppresses wordlines of those rows. Thus, the circuit 610 avoids unnecessary power consumption for refreshing row which are defined to not require to be refreshed by the minimum row address register 612. The circuit 610 can be altered in such a way that it either checks the minimum row address register 612 1) only in case of self-refresh mode or 2) both in self-refresh and auto-refresh mode.

[0065]As shown in FIG. 7, two parallel processes performed by decoder 616 are controlling the self-refresh - and with respective implementation also autorefresh - process 700. The two processes are performed by a global control circuit that includes the command decoder 616. The sub-process 702 controls the contents of the minimum row address register 612.

[0066]As shown in FIG. 7, the self-refresh process 700 involves at power up (step 704) resetting the value ROW_{min} stored in the minimum-row address register 612 to the maximum row address value per step 706. Next, the row-addresses of commands sent to the memory banks 208 are analyzed in the manner set forth below. In particular, detection of an extended mode register access for the minimum row address register 612 is performed per step 708.

If extended mode register access is detected per step 708, then the value of ROW_{min} is reset to the value of the maximum row-address (step 710) or alternatively programmed to any other value (not illustrated) so as to declare the contents of the particular bank in question irrelevant. If an extended mode register access is not detected per step 708, then the value of ROW_{min} is not altered.

[0067] Irrespective of whether or not an extended mode register access is detected per step 708, the presence of a write command is determined per step 712. If a write command is detected, then the row-address i gets decoded by command decoder 216 per step 714. If no write command is decoded then the process returns to step 708, as shown in FIG. 7, where another command sent to the memory banks 208 is monitored.

[0068] If a write command is detected, a comparison between the decoded value i and the value of ROW_{min} is performed per step 716. In particular, if the decoded value i of the row-address is less than the value ROW_{min}, the value ROW_{min} is set to the decoded value i and stored in the minimum row address register per step 718. Then the process is repeated starting at step 708 where another command sent to the memory banks 208 is monitored.

[0069] Note that if the decoded value i is greater than or equal to the value ROW_{min}, per step 716, then the above-described process is repeated at step 708 where another command sent to the memory banks 208 is monitored.

[0070] The subprocess 720 shows the behavior during self-refresh and autorefresh requests. In particular, the command decoder 616 detects whenever a self-refresh or auto-refresh request for any row-address occurs

per step 722. If no self-refresh or autorefresh is detected per step 722, then the process is then repeated and returns to step 722. If a self-refresh or autorefresh request is detected, then the value of the requested row-address for the refresh is compared with the value of ROW_{min} per step 724.

[0071] If the value of the requested row-address is higher or equal to the value ROW_{min}, then the rows of all banks 208 with this row-address are refreshed per step 726. The process is then repeated and returns to step 722.

[0072] Note that should the value of the requested row-address be determined in step 324 to be lower than the value ROW_{min} ,then the refresh request is ignored and the process returns to step 322, which results in lower power consumption for the refresh.

[0073] A second alternative variation of the self-refresh memory device and corresponding self-refresh process of FIGS. 2-3 is shown in FIGS. 8-9, wherein components and processes similar to those shown in FIGS. 2-3 and FIGS. 6-7 are identified with like numerals. As shown in FIG. 8, the memory device 804 includes both a maximum row address register 212 and a minimum row address register 612 that are in communication with both the command decoder 216 and the self-refresh circuit 210.

[0074] The command decoder monitors all write commands directed to the memory banks 208 and controls the contents of the maximum row address register 212 and the minimum row address register 612. The digital values of ROW_{max} and ROW_{min} of the maximum row address register 212 and the minimum row address register 612 are initially set to zero and a maximum row

address value, respectively, during the power-up sequence of the memory device 804. Afterwards, the values of ROW_{max} and ROW_{min} are controlled by the command decoder 816 of the memory device 804. Whenever a write-command is issued to the memory device 804, the command decoder 816 decodes this write command and also decodes the row-address to which data are written. Whenever the decoded row-address is lower than the value of ROW_{min} , then the value of ROW_{min} is updated to have a value that corresponds to the presently decoded row-address. Similarly, whenever the decoded row-address is higher than the value of ROW_{max} , then the value of ROW_{max} is updated to have a value that corresponds to the presently decoded row-address.

[0075] A command sequence– usually defined as an extended mode register set can be used to program/reset the values of ROW_{min} and ROW_{max} in a manner similar to that described with respect to the embodiments of FIGS. 2-3 and 6-7.

[0076] The self-refresh circuit 810 monitors the contents of the maximum row address register 212 and the minimum row address register 612 and starts the activate/precharge sequence only for those rows with an address that has a value that is less than or equal to the stored value of ROW_{min} and greater than or equal to the stored value of ROW_{max}. For all other rows, the self-refresh circuit 810 suppresses wordlines of those rows. Thus, the circuit 810 avoids unnecessary power consumption for refreshing row which are defined to not require to be refreshed by the minimum row address register 612 and the maximum row address register 212. The circuit 810 can be altered in

such a way that it either checks the maximum row address register 212 and the minimum row address register 612 1) only in case of self-refresh mode or 2) both in self-refresh and auto-refresh mode.

[0077] As shown in FIG. 9, two parallel processes are controlling the self-refresh - and with respective implementation also autorefresh - process 900. The two processes are performed by a global control circuit that includes the command decoder 816. The sub-process 902 controls the contents of both the maximum row address register 212 and the minimum row address register 612. The self-refresh process 900 involves at power up (step 904) resetting the values ROW_{max} and ROW_{min} to zero and the maximum row address value, respectively, per step 906. Next, the row-addresses of commands sent to the memory banks 208 are analyzed in the manner set forth below. In particular, detection of an extended mode register access for the maximum row address register 212 and the minimum row address register 612 is performed per step 908. If extended mode register access is detected per step 908, then the value of ROW_{max} is reset to zero or alternatively programmed to any other value (not illustrated) or ROW_{min} is reset to the value of the maximum row-address (step 910) or alternatively programmed to any other value (not illustrated) depending on whether the extended mode register access is detected to address the register 212 or register 612 so as to declare the contents of the particular bank in question irrelevant. If an extended mode register access is not detected per step 908, then the values of ROW_{max} and ROW_{min} are not altered.

[0078] Irrespective of whether or not an extended mode register access is detected per step 908, the presence of a write command is determined per step 912. If a write command is detected, then the row-address i gets decoded by command decoder 816 per step 914. If no write command is decoded then the process returns to step 908, as shown in FIG. 9, where another command sent to the memory banks 208 is monitored.

[0079] If a write command is detected, a comparison between the decoded value i and the values of ROW_{max} and ROW_{min} is performed in parallel per steps 916, 917. In particular, if the decoded value i of the row-address is greater than the value ROW_{max}, the value ROW_{max} is set to the decoded value i and stored in the maximum row address register per step 918. Similarly, if the decoded value i of the row-address is less than the value ROW_{min}, the value ROW_{min} is set to the decoded value i and stored in the minimum row address register per step 919. The process is repeated starting at step 908 where another command sent to the memory banks 208 is monitored.

[0080] Note that that while the addresses monitored at step 908 are not done sequentially, the addresses can be analyzed for each memory bank 208.

[0081] The subprocess 920 shows the behavior during self-refresh and autorefresh requests. In particular, the command decoder 816 detects whenever a self-refresh or auto-refresh request for any row-address occurs per step 922. If no self-refresh or autorefresh is detected per step 922, then the process is then repeated and returns to step 922. If a self-refresh or autorefresh request is detected, then the value of the requested row-address

for the refresh is compared with the values of ROW_{max} and ROW_{min} per step 924.

[0082] If the value of the requested row-address is lower or equal to the value ROW_{max} and higher or equal to the value ROW_{min} , then the rows of all banks 208 with this row-address get refreshed per step 926. The process is then repeated and returns to step 922.

[0083] Note that should the value of the requested row-address be determined in step 924 to be either higher than ROW_{max} or lower than the value ROW_{min}, then the refresh request gets ignored and the process returns to step 922, which results in lower power consumption for the refresh.

[0084] An alternative variation of the self-refresh memory device and corresponding self-refresh process of FIGS. 4-5 are shown in FIGS. 10-11, wherein components and processes similar to those shown in FIGS. 4-5 are identified with like numerals. As shown in FIG. 10, a memory system 1000 includes a memory controller 202 and a memory device 1004. The memory controller 202 has a normal refresh circuit 206 that performs a normal refresh operation in a manner similar to that described previously as explained previously with respect to FIGS. 4-5.

[0085] The memory controller 202 and the memory device 1004 are connected together by a bus 205 of command signals, a bus 207 of address signals, a bus 209 of data signals, clock signals (not illustrated) and datastrobe signals (not illustrated) in a manner similar to that described previously with respect to FIGS. 4-5.

[0086] The memory device 1004 has memory banks 208. Like the memory device 404 in FIG. 4, the memory device 1004 includes one address register per memory bank 208. Unlike memory device 404, the register is a minimum row address register 1012. The memory device 1004 also includes refresh enable circuits 1018, one for each memory bank 208.

[0087] The command decoder 1016 monitors all write commands directed to the memory banks 208 and controls the contents of the minimum row address registers 1012. The contents of the minimum row address register, connected to bank b will be denoted as $ROW_{min,b}$, a digital value which can cover the same range as the row-address space of the memory device.

[0088] The contents $ROW_{min,\,b}$ (b=1, 2, . . . n) of all minimum row address are set to a maximum row address value at the power-up sequence of the memory device 1004. Afterwards, the contents of the minimum row address registers 1012 are controlled by the command decoder 1016 of the memory device 1004. Whenever a write-command is issued to the memory device 1004, the command decoder 1016 decodes this write command and also decodes the bank address b and the row-address to which data are written. The decoder 1016 then compares the value $ROW_{min.\,b}$ of the minimum row address register corresponding to the bank b and the row-address of the decoded write command. If the row-address of the decoded write command is less than the value $ROW_{min,\,b}$ of the minimum row address register 1012 of the addressed bank b, the value $ROW_{min,\,b}$ of the minimum row address register 1012 for bank b is changed to value of the row-address of the decoded write command. A command sequence – usually defined as an

extended mode register set - can be used to reset the minimum row address registers 1012 of all banks 208 to a maximum address value. Alternatively, an implementation is possible where each minimum row address register is reset individually with an extended mode register. Another implementation can allow setting the contents of all minimum row address registers to a programmable value. Another implementation allows setting the contents of each minimum row address register to a programmable value individually.

[0089] The self-refresh circuit 1010 shown in FIG. 10 sends refresh requests, including the row-address information, to refresh enable circuits 1018 for all banks 208. These refresh enable circuits 1018 monitor the contents of the minimum row address registers for a corresponding memory bank 208. Only if the row-address of the refresh request from the self-refresh circuit 1010 is greater or equal to the value of $ROW_{min,b}$ of the minimum row-address register 1012 of a bank b, a refresh sequence is started for bank b. The circuit 1010 can be altered in such a way that it either does check the minimum row address register only in case of self-refresh mode or both in self-refresh and auto-refresh mode.

[0090] As shown in FIG. 11, the self-refresh process 1100 involves at power up (step 1104) resetting the value $ROW_{min,\ b}$ stored in the minimum-row address registers 1012 for each memory bank b (b=1, 2, . . . n) to a maximum address value per step 1106. Next, the row-addresses of commands sent to the memory banks 208 are analyzed in the manner set forth below. In particular, detection of an extended mode register access for each of the minimum row address registers 1012 is performed per step 1108. If extended

mode register access for a particular bank b is detected per step 1108, then the value of $ROW_{min, b}$ of the minimum row address register 412 for addressed bank b is reset to the maximum address value (step 1110) or alternatively programmed to any other value (not illustrated) so as to declare the contents of the particular bank in question irrelevant. If an extended mode register access is not detected per step 1108, then the value of $ROW_{min, b}$ is not altered.

[0091] Regardless of whether or not an extended mode register access is detected per step 1108, the presence of a write command is determined per step 1112. If a write command is detected, then the row-address i and addressed bank b gets decoded by command decoder 1016 per step 1114. If no write command is decoded then the process returns to step 1108, as shown in FIG. 11, where another command sent to the memory banks 208 is monitored.

[0092] If a write command is detected, the bank address b is decoded and a comparison between the decoded value of the row-address i and the value of $ROW_{min, b}$ is performed per step 1116. In particular, if the decoded value i of the row-address is less than the value $ROW_{min, b}$, the value $ROW_{min, b}$ is set to the decoded value i and stored in the minimum row address register 1012 corresponding to bank b per step 1118. Then the process is repeated starting at step 1108 where another command sent to the memory banks 208 is monitored.

[0093] Note that if the decoded value i is greater than or equal to the value

$ROW_{min, b}$, per step 1116, then the above-described process is repeated at step 1108 where another command sent to the memory banks 208 is monitored.

[0094] The subprocess 1120 shows the behavior during self-refresh and autorefresh requests. In particular, the command decoder 1016 detects whenever a self-refresh or auto-refresh request for any row-address of a particular bank b occurs per step 1122. If no self-refresh or autorefresh is detected per step 1122, then the process is then repeated and returns to step 1122. If a self-refresh or autorefresh request is detected, then several parallel steps are started in parallel, one for each bank 208. In particular, the value of the requested row-address for the refresh is compared by the refresh enable circuits 1018 in parallel with each of the values of $ROW_{min, b}$ of the banks 1, 2, . . . b per step 1124.

[0095] If the value of the requested row-address is greater than or equal to the value $ROW_{min, b}$ of the minimum row address register 1012 for bank b, then bank b gets refreshed at this row-address of the refresh request per step 1126. The process is then repeated and returns to step 1122.

[0096] Note that should the value of the requested row-address be determined in step 1124 to be lower than the value $ROW_{min, b}$,then the refresh request for bank b of the requested row-address gets ignored and the process returns to step 1122, which results in lower power consumption for the refresh.

[0097] Another alternative variation of the self-refresh memory device and corresponding self-refresh process of FIGS. 4-5 is shown in FIGS. 12-13,

wherein components and processes similar to those shown in FIGS. 4-5 are identified with like numerals. As shown in FIG. 12, a memory system 1200 includes a memory controller 202 and a memory device 1204. The memory controller 202 has a normal refresh circuit 206 that performs a normal refresh operation in a manner similar to that described previously as explained previously with respect to FIGS. 4-5.

[0098] The memory controller 202 and the memory device 1204 are connected together by a bus 205 of command signals, a bus 207 of address signals, a bus 209 of data signals, clock signals (not illustrated) and datastrobe signals (not illustrated) in a manner similar to that described previously with respect to FIGS. 4-5.

[0099] The memory device 1204 has memory banks 208. Like the memory device 404 in FIG. 4, the memory device 1004 includes one maximum address register 412 per memory bank 208. In addition, memory device 1004 includes one minimum address register 1012 per memory bank 208. The memory device 1204 also includes refresh enable circuits 1218, one for each memory bank 208.

[0100] The command decoder 1216 monitors all write commands directed to the memory banks 208 and controls the contents of the minimum row address registers 1012 and the maximum row address registers 412. The contents of the minimum and maximum row address registers, connected to bank b will be denoted as $ROW_{min,b}$, and $ROW_{max,\ b}$, respectively, digital values which can cover the same range as the row-address space of the memory device.

[0101] At power-up, the contents $ROW_{min,\,b}$ (b=1, 2, . . . n) of all minimum row address registers are set to a maximum row address value. Similarly, the contents $ROW_{max,\,b}$ (b=1, 2, . . . n) of all maximum row address registers are set to zero at the power-up sequence of the memory device 1204. Afterwards, the contents of the minimum row address registers 1012 and the maximum row address registers 412 are controlled by the command decoder 1216 of the memory device 1204. Whenever a write-command is issued to the memory device 1204, the command decoder 1216 decodes this write command and also decodes the bank address b and the row-address to which data are written. The decoder 1216 then compares the values $ROW_{min,b}$ and $ROW_{max,b}$ and the row-address of the decoded write command. If the row-address of the decoded write command is greater than the value $ROW_{max,b}$ of the addressed bank b, the value $ROW_{max,b}$ for bank b is changed to the value of the row-address of the decoded write command. If the row-address of the decoded write command is less than the value $ROW_{min,\,b}$ of the addressed bank b, the value $ROW_{min,\,b}$ for bank b is changed to value of the row-address of the decoded write command. A command sequence – usually defined as an extended mode register set - can be used to reset the minimum row address registers 1012 of all banks 208 to a maximum address value or alternatively programmed to any value. Similarly, the maximum row address registers 1012 of all banks 208 can be reset to zero or alternatively programmed to any value. Alternatively, an implementation is possible where each minimum row address register and each maximum row address register

is reset individually or programmed individually to any value with an extended mode register set.

[0102] The self-refresh circuit 1210 shown in FIG. 12 sends refresh requests, including the row-address information, to refresh enable circuits 1218 for all banks 208. These refresh enable circuits 1218 monitor the contents of the minimum and maximum row address registers for a corresponding memory bank 208. Only if the row-address of the refresh request from the self-refresh circuit 1210 is greater than or equal to the value of $ROW_{min,b}$ and less than or equal to the value of $ROW_{max,b}$ for the registers 412, 1012 of a bank b, a refresh sequence is started for bank b. The circuit 1210 can be altered in such a way that it either does check the minimum and maximum row address registers only in case of self-refresh mode or both in self-refresh and auto-refresh mode.

[0103] As shown in FIG. 13, the self-refresh process 1300 involves at power up (step 1304) resetting the value $ROW_{max,b}$ to zero and the value $ROW_{min, b}$ to a maximum address value, for the registers 412, 1012, respectively, for each memory bank b (b=1, 2, . . . n) per step 1306. Next, the row-addresses of commands sent to the memory banks 208 are analyzed in the manner set forth below. In particular, detection of an extended mode register access for each of the minimum and maximum row address registers is performed per step 1308. If extended mode register access for a particular bank b is detected per step 1308, then the value of $ROW_{max,b}$ is reset to zero or alternatively programmed to any other value (not illustrated) or $ROW_{min, b}$ is reset or alternatively programmed to any other value (not illustrated) to the

maximum address value (step 1310) depending on whether the extended mode register access is detected for register 412 or register 1012 so as to declare the content of the particular bank in question irrelevant. If an extended mode register access is not detected per step 1108, then the values of $ROW_{max,b}$ and $ROW_{min,b}$ are not altered.

[0104] Irrespective of whether or not an extended mode register access is detected per step 1308, the presence of a write command is determined per step 1312. If a write command is detected, then the row-address i and addressed bank b gets decoded by command decoder 1216 per step 1314. If no write command is decoded then the process returns to step 1308, as shown in FIG. 13, where another command sent to the memory banks 208 is monitored.

[0105] If a write command is detected, a comparison between the decoded value i and the values of $ROW_{max,b}$ and $ROW_{min,b}$ is performed per steps 1316 and 1317. In particular, if the decoded value i of the row-address is greater than the value $ROW_{max,b}$, the value $ROW_{max,b}$ is set to the decoded value i and stored in the maximum row address register 412 corresponding to bank b per step 1318. If the decoded value i of the row-address is less than the value $ROW_{min,b}$, the value $ROW_{min,b}$ is set to the decoded value i and stored in the minimum row address register 1012 corresponding to bank b per step 1319. Then the process is repeated starting at step 1308 where another command sent to the memory banks 208 is monitored.

[0106] The subprocess 1320 shows the behavior during self-refresh and autorefresh requests. In particular, the command decoder 1216 detects

whenever a self-refresh or auto-refresh request for any row-address of a particular bank b occurs per step 1322. If no self-refresh or autorefresh is detected per step 1322, then the process is then repeated and returns to step 1322. If a self-refresh or autorefresh request is detected, then the value of the requested row-address of the requested bank b for the refresh is compared with the values of $ROW_{max,b}$ and $ROW_{min, b}$ of the banks 1, 2, . . . b per step 1324.

[0107] If the value of the requested row-address of the requested bank b is lower than or equal to the value $ROW_{max,b}$ and greater than or equal to the value $ROW_{min, b}$ for bank b, then bank b gets refreshed at this row-address of the refresh request per step 1326. The process is then repeated and returns to step 1322.

[0108] Note that should the value of the requested row-address be determined in step 1324 to be either greater than greater than the value $ROW_{max,b}$ or less than the value $ROW_{min,b}$,then the refresh request for bank b of the requested row-address is ignored and the process returns to step 1322, which results in lower power consumption for the refresh.

[0109] Based on the above description of the processes shown in FIGS. 3, 5, 7, 9, 11 and 13, the design of the various components shown in FIGS. 2, 4, 6, 8, 10 and 12 based on existing DRAM products from vendors like Samsung, Micron, Elpida and Infineon is very straightforward for any DRAM designer or general logic designer.

[0110] The foregoing description is provided to illustrate the invention, and is not to be construed as a limitation. Numerous additions, substitutions and other changes can be made to the invention without departing from its scope as set forth in the appended claims.